

Worksheet 0

An echo of C++

No Marks are awarded for this worksheet, however, all tasks MUST be completed before preceding.

All work is handed in via git/GitLab and will require the skills demonstrated in this worksheet.

Compiling C++ with clang++

Before we begin it is important to be able to compile C++ programs, which you should already be confident at, but in case you are feeling rusty we compile a simple hello world program to get started. Assuming you are logged into the remote server, create a directory to work in and create the a C++ file (hello-world.cpp) with the following content:

```
#include <iostream>

int main(void) {

    std::cout << "Hello World" << std::endl;

    return 0;
}
```

This can be compiled with the following:

```
clang++ -o hello hello-world.cpp
```

and will result in the executable hello.

To add support for C++17 features, not used here but relied on later, simply add the command line option `-std=c++17`, so:

```
clang++ -std=c++17 -o hello hello-world.cpp
```

If you want to build directly from within VS Code you can add some `settings.json` and `tasks.json` to the directory `.vscode` at the top project directory. Versions of these files can be found on Blackboard, in the worksheets directory. Once installed you can build the project with `Ctrl-B` on Windows or Linux, and `Cmd-B` on Mac.

TicTacToe

In the introduction sessions and the first week's lecture we spent sometime looking at some basic C++ stuff and discussing implementing a very simple version of TicTacToe. For remaining of this worksheet spend sometime implementing elements of TicTacToe, in C++, on the remote server. Push your resulting implementation to Gitlab.

As noted in the lecture your program should use, at least:

- multiple source files (.cpp and .hpp);
- a class and corresponding object
- a lambda function

This should be a fairly straightforward task and we should aim to finish it ASAP.